PayPal™

Gateway Reporting Developer Guide

Last updated: Nov 2013

Gateway Reporting Developer Guide

Document Number: 200046.en_US-201311

© 1999 - 2013 PayPal, Inc. All rights reserved. PayPal is a registered trademark of PayPal, Inc. The PayPal logo is a trademark of PayPal, Inc. Other trademarks and brands are the property of their respective owners.
The information in this document belongs to PayPal, Inc. It may not be used, reproduced or disclosed without the written approval of PayPal, Inc.
Copyright © PayPal. All rights reserved. PayPal (Europe) S.à r.l. et Cie, S.C.A., Société en Commandite par Actions. Registered office: 22-24 Boulevard Royal, L-2449, Luxembourg, R.C.S. Luxembourg B 118 349
Consumer advisory: The PayPal™ payment service is regarded as a stored value facility under Singapore law. As such, it does not require the approval of the Monetary Authority of Singapore. You are advised to read the terms and conditions carefully.

Notice of non-liability:
PayPal, Inc. is providing the information in this document to you “AS-IS” with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice.

Contents

Preface 7

Audience 7
Organization 7
Where to Go for More Information 8
How to Contact Customer Service 8
Revision History 8

Chapter 1 Reporting Overview 11

Operations You Can Perform Using the Reporting API 11
Managing Report Templates 11
Running Reports by Report Template Name or Report Name 12
Obtaining Information about Reports 12
Managing Report Template Schedules 12
Performing Searches 12
URL for Running Applications 12
Live Transactions 12
Test Transactions 12

Chapter 2 Using the Reporting API 13

API Notation in this Chapter 13
Request Authorization Parameter Data 13
Response Data 14
Where to Go For More Information on the APIs 14
Running Reports 14
Running a Report and Obtaining Report Data 15
Running Reports Offline 16
Managing Report Templates 16
Using Report Templates 17
Running a report template and obtaining the report template data 17
Scheduling a Report Template 18
Managing Report Template Schedules 19
Performing Searches 20

Chapter 3 Response Codes and Status Codes 21

Response Codes 21
Status Codes 22

Appendix A Reporting XML Schema 23

Appendix B Reporting Samples 31

Running a Daily Activity Report 31
Running a Daily Activity Report Request 31
Running a Daily Activity Report Response 32
Running a Transaction Summary Report 33
Running a Transaction Summary Report Request 33
Running a Transaction Summary Report Response 34
Getting Results by Report ID 34
Getting Results by Report ID Request 34
Getting Results by Report ID Response 35
Getting Metadata 35
Getting Metadata Request 35
Getting Metadata Response 36
Format Returned 37
Getting Data 38
Getting Data Request 38
Getting Data Response 38
Data Returned 48
Creating a Report Template 49
Creating a Report Template Request 49
Creating a Report Template Response 49
Updating a Report Template 50
Updating a Report Template Request 50
Updating a Report Template Response 50
Getting a Report Template 51
Getting a Report Template Request 51
Getting a Report Template Response 51
Deleting a Report Template 52
Deleting a Report Template Request 52
Deleting a Report Template Response 52
Creating a Schedule 53
Creating a Schedule Request 53

Creating a Schedule Response . . . 53
Getting a Schedule . . . 54
Getting a Schedule Request . . . 54
Getting a Schedule Response . . . 54
Updating a Schedule . . . 55
Updating a Schedule Request . . . 55
Updating a Schedule Response . . . 55
Deleting a Schedule . . . 56
Deleting a Schedule Request . . . 56
Deleting a Schedule Response . . . 56
Running a Transaction ID Search . . . 57
Running a Transaction ID Search Request . . . 57
Running a Transaction ID Search Response . . . 57
Running a Profile - Next Payment Date Report . . . 58
Running a Profile - Next Payment Request . . . 58
Running a Profile - Next Payment Response . . . 58
Running a Profile - Profile Modified Date Report . . . 59
Running a Profile - Profile Modified Date Request . . . 59
Running a Profile - Profile Modified Date Response . . . 59
Retrieving a List of All Scheduled Reports that were Run . . . 60
Retrieving a List of All Scheduled Reports Request . . . 60
Retrieving a List of All Scheduled Reports Response . . . 60

Appendix C Report Parameters . . . 61

Reports in This Appendix . . . 61
ACH Activity Summary Report . . . 62
ACH Batch ID Report . . . 64
ACH Detail Return Activity Report . . . 66
ACH Notification of Change Report . . . 67
ACH Return Activity Report . . . 68
ACH Settlement Report . . . 69
ACH Transaction Report . . . 70
Auto Credit Report . . . 71
Auto Delayed Capture Report . . . 72
Auto Resettle Report . . . 74
Auto Void Report . . . 75
Batch ID Report . . . 76

Custom Report 77
Daily Activity Report 85
Filter Scorecard Report 87
Fraud Protection Report 88
Payflow Link Orders Report 89
PayPal Settlement Summary Report 91
Profile Next Payment Report 92
Profile Modified Date Report 93
Recurring Billing Report 94
Recurring Profile Failed Report 95
Recurring Profile Report 96
Settlement Report 97
Shipping and Billing Report 99
Transaction Summary Report 101

Appendix D Search Parameters 103

Transaction ID Search 103
Batch ID Search 105
AccountNumberSearch 106
CommentSearch 107
AccountNumberRefSearch 109
PurchaseOrderSearch 110
RecurringBillingProfileIDSearch 111
RecurringBillingProfileNameSearch 112
RecurringBillingAccountNumberSearch 113
RecurringBillingCommentSearch 114
RecurringBillingAmountSearch 115
FraudTransactionSearch 116
OrderIDSearch 118
PayPalTransactionIDSearch 120
PayPalEmailSearch 121
PayPalNameSearch 122

Index 123

Preface

Gateway Reporting Developer Guide describes the Reporting XML API for the following Gateway payment solutions:PayPal Payments Advanced, PayPal Payments Pro, Payflow Pro, and Payflow Link. The XML API enables you to develop Web and desktop applications that create, schedule, and run standard (pre-defined) and custom Payflow services reports.

Audience

This guide assumes that its readers:

- Have a good understanding of XML application development
- Have a background in payments services

Organization

This guide is organized as follows:

- Chapter 1, “Reporting Overview,” provides an overview of Reporting and the types of functionality it offers.
- Chapter 2, “Using the Reporting API,” describes how to use the Reporting API.
- Chapter 3, “Response Codes and Status Codes,” identifies all supported standard reports, along with the valid and required parameters of each. It also presents status codes and response codes, along with their associated message strings.
- Appendix A, “Reporting XML Schema,” provides the XML schema upon which the Reporting XML API is based.
- Appendix B, “Reporting Samples,” presents samples demonstrating how to use the reporting API.
- Appendix C, “Report Parameters,” describes the input and output parameters for each standard report.
- Appendix D, “Search Parameters,” describes the input and output parameters for performing searches by search name.

Where to Go for More Information

PayPal Manager online help describes the use of PayPal Manager—the web-based administration tool that you use to process transactions manually, issue credits, and generate reports.

How to Contact Customer Service

For answers to specific questions about PayPal products, contact Customer Service.

Revision History

Revision history for *Gateway Reporting Developer Guide*.

TABLE P.1 Revision History

Date	Description
November 2013	Added a note on how to retrive all report templates in Getting a Report Template Request. Added a note on how to retrieve all schedules in Getting a Schedule Request. Added the sample report: Retrieving a List of All Scheduled Reports that were Run.
May 2012	Updated reportId description - reportId is valid for up to 120 days.
February 2012	Added 4 Settlement report response parameters.
January 2012	Added 7 ACH reports; edits for technical accuracy.
December 2011	Updated for technical accuracy.
August 2011	Added a comment that the last 4 digits of the account number are returned in the Account Number output field.
June 2011	Added two new reports: Profile - Profile Modified Date and ProfileNext Payment Date.
June 2010	Updated for technical accuracy.
December 2009	Added Fraud Transaction Search table.
November 2008	Updated links to test and live reporting URLs.
June 2008	Minor clarifications and corrections.
August 2007	Updated Title page and logo.
July 2006	Updated document title, product names. Reformatted in PayPal templates. Updated to support new currencies.

TABLE P.1 Revision History

Date	Description
March 2006	Integrated Express Checkout feature.

1 Reporting Overview

The Reporting API is an API that you can use to automate your report queries. It allows you to programmatically query in XML the Reporting database.

PayPal Manager is also a client of the Reporting API. Everything that Manager can do with reports is available to you through API calls. You can, for example, request a particular report be run within a specified date range. The response returns all the data that the report generates. The reporting engine encapsulates the information in all the Gateway services reports. You can store this information in your local database and use it as needed.

Operations You Can Perform Using the Reporting API

The Reporting API enables you to perform the following operations:

- Managing report templates
- Running reports on demand by report or report template name
- Obtaining information about reports
- Managing report template schedules
- Performing searches

The *standard* (predefined) reports supported are described in Appendix C, “Report Parameters.”

Managing Report Templates

You may want to create report templates based on the reports that are supported. A report template lets you create an instance of a report. With this instance, you provide a list of parameters.

Report templates are handy when you find that you frequently need to look up reports based on the same criteria. You provide the parameters and values once, and Reporting saves this information with the report template. When you run the report template, you do not need to provide the parameters a second time.

Report template management enables you to perform all the following tasks:

- Create report templates
- Query report templates in the database
- Retrieve report templates from the database
- Update report templates
- Delete report templates

Running Reports by Report Template Name or Report Name

You can run reports and report templates. To run a report or report template, you submit a `runReportRequest`, passing in the respective report name or report template name. You can also pass in additional parameters in with a request.

To submit a request to the Reporting API, you post your XML request via HTTPS to the host URLs in the body of the request. No SDK is needed since you just perform a standard HTTPS post using the language of your choice. The Content-Type for the request should be text/plain.

Obtaining Information about Reports

You can obtain the following information about reports:

- Report status
- Report results

Managing Report Template Schedules

You can create, update, and delete report template schedules.

Performing Searches

You can search for payment transaction data based on search names such as TransactionIDSearch, BatchIDSearch, and AccountNumberSearch.

URL for Running Applications

Use the following URLs for running Reporting applications.

Live Transactions

https://payments-reports.paypal.com/reportingengine

Test Transactions

https://payments-reports.paypal.com/test-reportingengine

2 Using the Reporting API

This chapter describes how you can use the Reporting API to perform the following tasks.

- "Running Reports" on page 14
- "Managing Report Templates" on page 16
- "Using Report Templates" on page 17
- "Performing Searches" on page 20

API Notation in this Chapter

The chapter uses the following short-hand notation for referring to Reporting API requests and parameter information.

requestName (paramName, paramName, [...])

requestName is the name of the request. The request input parameters, if any, are listed in a parameter list enclosed in parentheses.

Request Authorization Parameter Data

The Reporting API notation in this chapter does not include the authorization information that is passed with every request. All Reporting API requests require that you specify these parameters and values as your user information.

TABLE 2.1 User information

Parameter	Description
user	If you set up one or more additional users on the account, this value is the ID of the user authorized to process transactions. If, however, you have not set up additional users on the account, user has the same value as vendor.
vendor	Your merchant login ID that you created when you registered for the Payflow Pro account.
partner	The ID provided to you by the authorized PayPal Reseller who registered you for the Payflow Pro service. If you purchased your account directly from PayPal, use PayPal.
password	The 6- to 32-character password that you defined while registering for the account.

Response Data

Responses include the following information:

- All responses return a base response consisting of a response code (`responseCode`) and a response message (`responseMsg`).
- Most responses return applicable response parameter data.
- Some responses return a status code (`statusCode`) and a status message (`statusMsg`).

Where to Go For More Information on the APIs

To understand how the XML API is coded, look at the request notation along with the XML schema, samples, and report parameters in this guide. See the following sources for details:

Source	Contents
Chapter 3, "Response Codes and Status Codes"	Lists all response codes, status codes, and their message strings.
Appendix A	Contains the XML schema.
Appendix B	Provides API request and response samples for each request and response referred to in this chapter.
Appendix C	Describes the input and output parameters for all *standard* (predefined) reports supported.
Appendix D	Describes the input and output parameters for all report searches supported.

Running Reports

You can run reports and obtain the report data right away. If, however, running a report takes more than the time allotted, the report is run offline, and you must periodically check for the report data.

Running a Report and Obtaining Report Data

This procedure describes how to run a report and how to obtain the data in that report.

Step 1. Run the report

To run a report, call `runReportRequest`, providing the following parameter information.

```
runReportRequest(reportName, reportParams, pageSize)
```

Parameter	Description
reportName	Name of the report, for example, DailyActivityReport.
reportParam(s)	(Optional) paramName and paramValue for each request input param to be assigned a value.
pageSize	Number of rows of data per page. Default is 50.

The `runReportResponse` returns the `reportId`. The `reportId` is used to refer to the particular report in future requests. The `reportId` returned is valid for up to 120 days.

Step 2. Get the metadata

To get information on how the report is formatted, call `getMetaDataRequest`, providing the `ReportId` returned when the report was run.

```
getMetaDataRequest(ReportId)
```

`getMetaDataResponse` returns the following data.

Parameter	Description
numberOfRows	Total number of rows.
numberOfPages	Total number of pages in the report.
pageSize	Page size.
numberOfColumns	Total number of columns.
dataName	Name of each column.
dataType	Type of column data, for example, string.

Step 3. Get the report data

To get the actual data in the report, call `getDataRequest` with the following parameters.

```
getDataRequest(reportId, pageNum)
```

reportId	ID for this report returned by runReportRequest; `reportId` is valid for up to 120 days.
pageNum	Report page number containing data to be returned.

Data is returned for the report page specified by `pageNum`. To obtain the data in a multi-page report, you must call `getDataRequest` for each report page.

Running Reports Offline

When a report takes longer to run than the time allotted, it is taken offline and status code 2, "Report is currently executing," is returned. You must check for the results at a later time.

Step 1. Run the report

Request to run a report by calling `runReportRequest`. See "Step 1. Run the report" on page 15.

The report times out. The `ReportId` and status code 2 are returned.

Step 2. Get the report results later

It is your responsibility to obtain the report results at a later time. Call `getResultsRequest`, providing the `reportId` as a parameter.

```
getResultsRequest(reportId)
```

`getResultsResponse` returns a `statusCode` and `statusMsg`. Call `getResultsResponse` in a loop until `statusCode` 3, "Report has completed successfully," is returned.

Step 3. Get the metadata

Request to get the report's metadata by calling `getMetaDataRequest`. See "Step 2. Get the metadata" on page 15.

Step 3. Get the report data

Request to get the report data by calling `getDataRequest`. See "Step 3. Get the report data" on page 15.

Managing Report Templates

Using the Reporting API, you can create, update, and delete report templates.

Step 1. Create a report template

To create a report template, call `createTemplateRequest` with the following parameters.

```
createTemplateRequest (templateName, reportName,
reportParams)
```

Parameter	Description
templateName	Name that you are giving to the report template.
reportName	Standard report from which you are creating the report template.

reportParam(s)	paramName and paramValue of each reportParam to be assigned a value.

The response to creating a report template returns a `respCode` and `respMsg`.

Step 2. Update a report template

To update a report template, call the `updateTemplateRequest` with the following parameters.

```
updateTemplateRequest(templateName, reportParams)
```

templateName	Name given to the report template when it was created.
reportParam(s)	paramName and paramValue of each reportParam to be assigned a value.

The response to updating a report template returns a `respCode` and `respMsg`.

NOTE: Updating a report template is not incremental. It only sets values for the parameters that are passed to `updateTemplateRequest`.

Step 3. Delete a report template

To delete a report template, call `deleteTemplateRequest`, providing the `templateName` as the input parameter.

```
deleteTemplateRequest(templateName)
```

The response to deleting a report template returns a `respCode` and `respMsg`.

Using Report Templates

Report templates are run like any standard report. They can also be scheduled to run on a daily, weekly, or monthly basis. Finally, report template schedules can be updated or deleted.

Running a report template and obtaining the report template data

Step 1. Run a report template

To run a report template, call `runReportRequest` with the following parameters.

```
runReportRequest(templateName, reportParams, pageSize).
```

templateName	Name given to the report template when it was created.

reportParam(s)	(Optional) paramName and paramValue for each reportParam that you want to assign a value to. These parameters can be ones previously saved in the report template or additional ones. The parameter values take precedence over the values previously saved for the parameters in the report template.
pageSize	Number of rows of data per page. Default is 50.

Step 2. Get the metadata

Request to get the report's metadata by calling `getMetaDataRequest`. See "Step 2. Get the metadata" on page 15.

Step 3. Get the report data

Request to get the report data by calling `getDataRequest`. See "Step 3. Get the report data" on page 15.

Scheduling a Report Template

You can schedule a report template to run on a daily, weekly, or monthly basis. The Reporting API includes functionality to create schedules and to get the results of a scheduled report template.

Step 1. Create a schedule

To create a schedule for a report template, call `createScheduleRequest` with the following parameters.

```
createScheduleRequest(scheduleName, templateName, schedule)
```

scheduleName	Name that you give to the schedule.
templateName	Name of the report template to be scheduled.
schedule	One of three values: • daily • weekly (select dayOfWeek value: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday) • monthly (Specify dayOfMonth value as an integer from 1 to 31. If you specify 30 when the month is 28 days long, the value is corrected to 28.) The time of the day starts at midnight the previous night. The value daily, for example, means midnight the previous night to midnight tonight.

Step 2. Get the results

To get the results of a scheduled report template that is run, call `getScheduleRequest` with `scheduleName` as a parameter.

```
getScheduleRequest(schedulename)
```

`getScheduleResponse` returns all the `reportIds` and execution dates of the reports that were run on the schedule.

Step 3. Get the metadata

For each report, call `getMetaDataRequest`, passing in the reportId as a parameter, to get the information on how that report is formatted. See "Step 2. Get the metadata" on page 15.

Step 4. Get the report data

Request to get the report data for each report that was run by calling `getDataRequest`. See "Step 3. Get the report data" on page 15.

Managing Report Template Schedules

Using the Reporting API, you can update and delete report template schedules.

Update a schedule

To update a schedule, call the `updateScheduleRequest` with the following parameters.

```
updateScheduleRequest(scheduleName, templateName, schedule)
```

scheduleName	Name given to the schedule when it was created.
templateName	Name of the report template to update. Use this parameter to specify the name of a different report template to associate with this schedule update.
schedule	Use this parameter to change the schedule. schedule is one of three values: • daily • weekly (select dayOfWeek value: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday) • monthly (specify dayOfMonth value as an integer from 1 to 31. (Specify dayOfMonth value as an integer from 1 to 31. If you specify 30 when the month is 28 days, the value is corrected to 28.) The time of the day starts at midnight the previous night. The value daily, for example, means midnight the previous night to midnight tonight.

The response to updating a schedule returns a `respCode` and `respMsg`.

Delete a schedule

To delete a schedule, call `deleteScheduleRequest`, providing the `scheduleName` as the input parameter.

```
deleteTemplateRequest(scheduleName)
```

The response to deleting a schedule returns a `respCode` and `respMsg`.

Performing Searches

Using the Reporting API, you can search for transaction data by any of the following search names.

- TransactionIDSearch
- BatchIDSearch
- AccountNumberSearch
- CommentSearch
- AccountNumberRefSearch
- PurchaseOrderSearch
- RecurringBillingProfileIDSearch
- RecurringBillingProfileNameSearch
- RecurringBillingAccountNumberSearch
- RecurringBillingCommentSearch
- RecurringBillingAmountSearch

Step 1. Run the search request

To run a search request, call `runSearchRequest` with the following parameters.

```
runSearchRequest(searchName, reportParam(s), pageSize)
```

searchName	One of the search names listed above, for example, TransactionIDSearch.
reportParam(s)	paramName and paramValue of one or more input parameters for this search. See Appendix C, "Report Parameters," for the required and optional parameters.
pageSize	Number of rows of data per page. Default is 50.

`runSearchResponse` returns the `reportId` for this search as well as the `statusCode` and `statusMsg`. The `reportId` returned is valid for up to 120 days.

Step 2. Get the metadata

Request to get the search metadata by calling `getMetaDataRequest`. See "Step 2. Get the metadata" on page 15.

Step 3. Get the report data

Request to get the search data by calling `getDataRequest`. See "Step 3. Get the report data" on page 15.

3 Response Codes and Status Codes

This chapter provides the following details on reports. It identifies:

- Response codes and response messages
- Status codes and the status messages

Response Codes

Response codes indicate the success or failure of a Reporting request. Table 5-2 describes the supported response codes and messages returned with each.

TABLE 3.1 Response codes and response messages

Response Code	Response Message
100	Request has completed successfully
101	Request has failed
102	An internal scheduler error has occurred
103	Unknown report requested
104	Invalid Report ID
105	A system error has occurred
106	A database error has occurred
107	Invalid XML request
108	User authentication failed
109	Invalid report parameters provided
110	Invalid merchant account
111	Invalid page number
112	Template already exists
113	Unknown template requested

Status Codes

Status codes indicate the status of a report, not necessarily the status of the Reporting request to be executed. A report may fail even if a Reporting request succeeds.

TABLE 3.2 Status codes and status messages

Status Code	Status Message
1	Report has been created
2	Report is currently executing
3	Report has completed successfully
4	Report has failed
5	Report has expired
6	Report has expired

A Reporting XML Schema

This appendix contains the XML Reporting 1.0 schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="reportingEngineRequest">
        <xs:complexType>
           <xs:sequence>
              <xs:element name="authRequest">
                 <xs:complexType>
                    <xs:sequence>
                       <xs:element name="user" type="xs:string"/>
                       <xs:element name="vendor" type="xs:string"/>
                       <xs:element name="partner" type="xs:string"/>
                       <xs:element name="password" type="xs:string"/>
                       <xs:element name="executor" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                 </xs:complexType>
              </xs:element>
              <xs:choice>
                 <xs:element name="createTemplateRequest">
                    <xs:complexType>
                       <xs:sequence>
                          <xs:element ref="templateDefinition"/>
                       </xs:sequence>
                    </xs:complexType>
                 </xs:element>
                 <xs:element name="getTemplateRequest">
                    <xs:complexType>
                       <xs:choice>
                          <xs:element ref="reportName"/>
                          <xs:element ref="templateName"/>
                       </xs:choice>
                    </xs:complexType>
                 </xs:element>
                 <xs:element name="updateTemplateRequest">
                    <xs:complexType>
                       <xs:sequence>
                          <xs:element ref="templateDefinition"/>
                       </xs:sequence>
                    </xs:complexType>
                 </xs:element>
                 <xs:element name="deleteTemplateRequest">
                    <xs:complexType>
                       <xs:sequence>
```

```
                        <xs:element ref="templateName"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="createScheduleRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="scheduleDefinition"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getScheduleRequest">
                <xs:complexType>
                    <xs:choice>
                        <xs:element ref="scheduleName"/>
                        <xs:element ref="templateName"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
            <xs:element name="updateScheduleRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="scheduleDefinition"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="deleteScheduleRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="scheduleName"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="runReportRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:choice>
                            <xs:element ref="reportName"/>
                            <xs:element ref="templateName"/>
                        </xs:choice>
                        <xs:element ref="reportParam" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element ref="notificationRequired" minOccurs="0"/>
                        <xs:element ref="pageSize" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="runSearchRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="searchName"/>
```

```
                        <xs:element ref="reportParam" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element ref="pageSize" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getResultsRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:choice minOccurs="0">
                            <xs:element ref="scheduleName"/>
                            <xs:element ref="templateName"/>
                            <xs:element ref="reportName"/>
                            <xs:element ref="searchName"/>
                            <xs:element ref="reportId"/>
                        </xs:choice>
                        <xs:choice minOccurs="0">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="startDate"type="xs:date"/>
                                    <xs:element name="endDate" type="xs:date"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:choice>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getMetaDataRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="reportId"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getDataRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="reportId"/>
                        <xs:element ref="pageNum" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="reportingEngineResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="baseResponse">
                <xs:complexType>
```

```
                        <xs:sequence>
                            <xs:element name="responseCode" type="xs:int"/>
                            <xs:element name="responseMsg" type="xs:string"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:choice minOccurs="0">
                    <xs:element name="getTemplateResponse">
                        <xs:complexType>
                            <xs:choice>
                                <xs:element ref="templateName" maxOccurs="unbounded"/>
                                <xs:element ref="templateDefinition"/>
                            </xs:choice>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="getScheduleResponse">
                        <xs:complexType>
                            <xs:choice>
                                <xs:element ref="scheduleName" maxOccurs="unbounded"/>
                                <xs:element ref="scheduleDefinition"/>
                            </xs:choice>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="runReportResponse">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="reportId"/>
                                <xs:element ref="statusCode"/>
                                <xs:element ref="statusMsg"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="runSearchResponse">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element ref="reportId"/>
                                <xs:element ref="statusCode"/>
                                <xs:element ref="statusMsg"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="getResultsResponse">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Results" minOccurs="0"
maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element ref="reportId"/>
                                            <xs:element ref="statusCode"/>
                                            <xs:element ref="statusMsg"/>
```

```
                                        <xs:element name="executionTime"
type="xs:dateTime" minOccurs="0"/>
                                        <xs:element ref="scheduleName" minOccurs="0"/>
                                        <xs:element ref="reportName" minOccurs="0"/>
                                        <xs:element ref="templateName" minOccurs="0"/>
                                        <xs:element ref="searchName" minOccurs="0"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="getMetaDataResponse">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="numberOfRows" type="xs:int"/>
                            <xs:element name="numberOfPages" type="xs:int"/>
                            <xs:element ref="pageSize"/>
                            <xs:element name="numberOfColumns" type="xs:int"/>
                            <xs:element name="columnMetaData" minOccurs="0"
maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="dataName" type="xs:string"/>
                                        <xs:element name="dataType">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:enumeration value="string"/>
                                                    <xs:enumeration value="number"/>
                                                    <xs:enumeration value="date"/>
                                                    <xs:enumeration value="currency"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="getDataResponse">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="reportDataRow" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="columnData"
maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:sequence>
```

```
                                                    <xs:element name="data"
type="xs:string" minOccurs="0"/>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element ref="pageNum"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="reportDefinition">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="reportId" type="xs:string"/>
<xs:element name="paramName" type="xs:string"/>
<xs:element name="paramValue" type="xs:string"/>
<xs:element name="templateName" type="xs:string"/>
<xs:element name="reportName" type="xs:string"/>
<xs:element name="reportParam">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="paramName"/>
            <xs:element ref="paramValue"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="templateDefinition">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="templateName"/>
            <xs:element ref="reportName"/>
            <xs:element ref="reportParam" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="notificationRequired" type="xs:boolean"/>
<xs:element name="scheduleDefinition">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="scheduleName"/>
            <xs:element ref="templateName"/>
            <xs:element ref="notificationRequired"/>
            <xs:choice>
```

```
                    <xs:element name="daily">
                        <xs:complexType/>
                    </xs:element>
                    <xs:element name="weekly">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="dayOfWeek">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="Sunday"/>
                                            <xs:enumeration value="Monday"/>
                                            <xs:enumeration value="Tuesday"/>
                                            <xs:enumeration value="Wednesday"/>
                                            <xs:enumeration value="Thursday"/>
                                            <xs:enumeration value="Friday"/>
                                            <xs:enumeration value="Saturday"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="monthly">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="dayOfMonth" type="xs:int"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:choice>
                <xs:element ref="pageSize" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="scheduleName" type="xs:string"/>
    <xs:element name="pageNum" type="xs:int"/>
    <xs:element name="searchName" type="xs:string"/>
    <xs:element name="pageSize" type="xs:int"/>
    <xs:element name="statusCode" type="xs:int"/>
    <xs:element name="statusMsg" type="xs:string"/>
</xs:schema>
```

A Reporting XML Schema

B Reporting Samples

This appendix demonstrates how to use the Reporting API to create, run, and manage reports. For details on the predefined reports named in this appendix, see the *Manager User's Guide*.

NOTE: In this appendix, replace these values shown in boldface with appropriate values: **User**, **Vendor**, **Partner**, and **Password**.

Running a Daily Activity Report

This section demonstrates how to run a Daily Activity Report.

Running a Daily Activity Report Request

The request demonstrates how to run a Daily Activity Report with a page size of 50 lines to obtain results for September 7, 2007.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <runReportRequest>
            <reportName>DailyActivityReport</reportName>
            <reportParam>
                <paramName>report_date</paramName>
               <paramValue>2007-09-07</paramValue>
            </reportParam>
            <pageSize>50</pageSize>
     </runReportRequest>
</reportingEngineRequest>
```

Running a Daily Activity Report Response

In the response to running the Daily Activity Report, the `responseCode` of 100 indicates that the report completed successfully. The `reportId` 132 returned is used to uniquely identify this report in future calls to the Reporting API.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
        <baseResponse>
            <responseCode>100</responseCode>
            <responseMsg>Request has completed successfully</responseMsg>
        </baseResponse>
        <runReportResponse>
            <reportId>132</reportId>
            <statusCode>3</statusCode>
            <statusMsg>Report has completed successfully</statusMsg>
        </runReportResponse>
</reportingEngineResponse>
```

Running a Transaction Summary Report

This sample demonstrates how to run a Transaction Summary Report.

Running a Transaction Summary Report Request

The request demonstrates how to run a Transaction Summary Report with a page size of 50 lines starting on September 7, 2007 at 12:00AM and ending on September 7, 2007 at 12:59 and 59 seconds PM.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <runReportRequest>
            <reportName>TransactionSummaryReport</reportName>
            <reportParam>
                <paramName>start_date</paramName>
               <paramValue>2007-09-07 00:00:00</paramValue>
            </reportParam>
            <reportParam>
                <paramName>end_date</paramName>
               <paramValue>2007-09-07 23:59:59</paramValue>
            </reportParam>
            <pageSize>50</pageSize>
     </runReportRequest>
</reportingEngineRequest>
```

Running a Transaction Summary Report Response

In the response to running the Transaction Summary Report, the response code of 100 indicates that the request completed successfully. The status code of 3 indicates that the report completed successfully and the `reportId` 131 is returned for future reverence in calls inquiring about this report.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
     <runReportResponse>
        <reportId>133</reportId>
        <statusCode>3</statusCode>
        <statusMsg>Report has completed successfully</statusMsg>
     </runReportResponse>
</reportingEngineResponse>
```

Getting Results by Report ID

This sample demonstrates how to retrieve the results of a report that was previously run.

Getting Results by Report ID Request

In the request to get the results by report ID, `reportId` 131 specifies the report for which results are being requested. The `reportId` 131 was returned in a previous call to run the report.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <getResultsRequest>
        <reportId>131</reportId>
     </getResultsRequest>
</reportingEngineRequest>
```

Getting Results by Report ID Response

In the response to getting results by report ID, the response code of 100 indicates that the request completed successfully. The status code of 3 indicates that the report with `reportId` 131 completed successfully.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
     <getResultsResponse>
        <Results>
           <reportId>131</reportId>
           <statusCode>3</statusCode>
           <statusMsg>Report has completed successfully</statusMsg>
        </Results>
     </getResultsResponse>
</reportingEngineResponse>
```

Getting Metadata

This sample demonstrates how to retrieve the format of the data in a previously run report.

Getting Metadata Request

The request demonstrates how to obtain the formatting information for the report with `reportId` 131.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <getMetaDataRequest>
        <reportId>131</reportId>
     </getMetaDataRequest>
</reportingEngineRequest>
```

Getting Metadata Response

In the response to retrieving metadata, the response code of 100 indicates that the request completed successfully. The response returns the number columns, rows, pages, and lines per page are present in the format of the report with `reportID` 131. For each table column, the response returns the column heading.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
     <getMetaDataResponse>
        <numberOfRows>13</numberOfRows>
        <numberOfPages>1</numberOfPages>
        <pageSize>50</pageSize>
        <numberOfColumns>11</numberOfColumns>
        <columnMetaData>
           <dataName>Transaction ID</dataName>
           <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Trans Time</dataName>
           <dataType>date</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Trans Type</dataName>
           <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Tender Type</dataName>
           <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Account Number</dataName>
           <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Expires</dataName>
           <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Amount</dataName>
           <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
           <dataName>Result</dataName>
           <dataType>number</dataType>
        </columnMetaData>
        <columnMetaData>
```

```
            <dataName>Reponse Msg</dataName>
            <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
            <dataName>Comment1</dataName>
            <dataType>string</dataType>
        </columnMetaData>
        <columnMetaData>
            <dataName>Comment2</dataName>
            <dataType>string</dataType>
        </columnMetaData>
    </getMetaDataResponse>
</reportingEngineResponse>
```

Format Returned

The format returned by this sample is shown below:

NOTE: The last four digits of the Account Number are returned.

Transaction ID	Trans Time	Trans Type	Tender Type	Account Number	Expires	Amount	Result	Response Msg	Comment1	Comment2

Getting Data

This sample demonstrates how to retrieve the data in a previously run report.

Getting Data Request

In the request to retrieve data, `reportId` 131 specifies the report for which the data is being requested. The `reportId` value of 131 was returned in a previous call to run the report. The `pageNum` refers to the page number desired in a multi-page report.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <getDataRequest>
        <reportId>131</reportId>
        <pageNum>1</pageNum>
     </getDataRequest>
</reportingEngineRequest>
```

Getting Data Response

In the response to retrieving data, the response code of 100 indicates that the request completed successfully. The response returns the data in each of the columns and rows of the report with `reportID` 131.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
     <getDataResponse>
        <reportDataRow>
           <columnData>
              <data>V89A0A000087</data>
           </columnData>
           <columnData>
              <data>2007-09-07 17:45:08</data>
           </columnData>
           <columnData>
              <data>S</data>
           </columnData>
           <columnData>
              <data>0</data>
```

```
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A000088</data>
        </columnData>
        <columnData>
            <data>2007-09-07 17:46:09</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
```

```
            <columnData>
                <data/>
            </columnData>
            <columnData>
                <data/>
            </columnData>
        </reportDataRow>
        <reportDataRow>
            <columnData>
                <data>V89A0A000089</data>
            </columnData>
            <columnData>
                <data>2007-09-07 17:48:20</data>
            </columnData>
            <columnData>
                <data>S</data>
            </columnData>
            <columnData>
                <data>0</data>
            </columnData>
            <columnData>
                <data>4111XXXXXXXX1111</data>
            </columnData>
            <columnData>
                <data>01/07</data>
            </columnData>
            <columnData>
                <data>A1000</data>
            </columnData>
            <columnData>
                <data>0</data>
            </columnData>
            <columnData>
                <data>Approved</data>
            </columnData>
            <columnData>
                <data/>
            </columnData>
            <columnData>
                <data/>
            </columnData>
        </reportDataRow>
        <reportDataRow>
            <columnData>
                <data>V89A0A00008A</data>
            </columnData>
            <columnData>
                <data>2007-09-07 17:52:03</data>
            </columnData>
            <columnData>
                <data>S</data>
```

```
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
            <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A00008B</data>
        </columnData>
        <columnData>
            <data>2007-09-07 17:53:07</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
```

```
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A00008C</data>
        </columnData><columnData>
            <data>2007-09-07 17:53:32</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        columnData>
            data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A00008D</data>
        </columnData>
        <columnData>
            <data>2007-09-07 17:54:20</data>
        </columnData>
```

```
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            <data>0</data>
            </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A00008E</data>
        </columnData>
        <columnData>
            <data>2007-09-07 17:56:02</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            data>0</data>
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
```

```
            <data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A00008F</data>
        </columnData>
        <columnData>
            <data>2007-09-07 17:56:33</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/07</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>Approved</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A000090</data>
        </columnData>
```

```
        <columnData>
            <data>2007-09-07 18:01:31</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>01/05</data>
        </columnData>
        <columnData>
            <data>A1000</data>
        </columnData>
        <columnData>
            <data>24</data>
        </columnData>
        <columnData>
            <data>Unknown error</data>
        </columnData>
        <columnData>
            <data/>
        </columnData>
        <columnData>
            <data/>
        </columnData>
    </reportDataRow>
    <reportDataRow>
        <columnData>
            <data>V89A0A000091</data>
        </columnData>
        <columnData>
            <data>2007-09-07 18:01:44</data>
        </columnData>
        <columnData>
            <data>S</data>
        </columnData>
        <columnData>
            <data>0</data>
        </columnData>
        <columnData>
            <data>4111XXXXXXXX1111</data>
        </columnData>
        <columnData>
            <data>12/05</data>
        </columnData>
        <columnData>
            <data>A1000</data>
```

```
            </columnData>
            <columnData>
                <data>0</data>
            </columnData>
            <columnData>
                <data>Approved</data>
            </columnData>
            <columnData>
                <data/>
            <columnData>
                <data/>
            </columnData>
         </reportDataRow>
        <reportDataRow>
            <columnData>
                <data>V89A0A000092</data>
            </columnData>
            <columnData>
                <data>2007-09-07 18:02:19</data>
            </columnData>
            <columnData>
                <data>S</data>
            </columnData>
            <columnData>
                <data>0</data>
            </columnData>
            <columnData>
                <data>4111XXXXXXXX1111</data>
            </columnData>
            <columnData>
                <data>12/05</data>
            </columnData>
            <columnData>
                <data>A1000</data>
            </columnData>
            <columnData>
                <data>0</data>
            </columnData>
            <columnData>
                <data>Approved</data>
            </columnData>
            <columnData>
                <data/>
            </columnData>
            <columnData>
                <data/>
            </columnData>
        </reportDataRow>
    <reportDataRow>
            <columnData>
                <data>V89A0A000093</data>
```

```
                </columnData>
                <columnData>
                    <data>2007-09-07 18:03:09</data>
                </columnData>
                <columnData>
                    <data>S</data>
                </columnData>
                <columnData>
                    <data>0</data>
                </columnData>
                <columnData>
                    <data>4111XXXXXXXX1111</data>
                </columnData>
                <columnData>
                    <data>12/05</data>
                </columnData>
                <columnData>
                    <data>A1000</data>
                </columnData>
                <columnData>
                    <data>0</data>
                </columnData>
                <columnData>
                    <data>Approved</data>
                </columnData>
                <columnData>
                    <data/>
                </columnData>
                <columnData>
                    <data/>
                </columnData>
            </reportDataRow>
            <pageNum>1</pageNum>
        </getDataResponse>
</reportingEngineResponse>
```

Data Returned

The data is returned in the following tabular format of 13 rows and 11 columns:

V89A0A000087	2007-09-07 17:45:07	S	O	1111	01/07	A1000	O	Approved		
V89A0A000088	2007-09-07 17:45:09	S	O	1111	01/07	A1000	O	Approved		
V89A0A000089	2007-09-07 17:48:20	S	O	1111	01/07	A1000	O	Approved		
V89A0A00008A	2007-09-07 17:52:03	S	O	1111	01/07	A1000	O	Approved		
V89A0A00008B	2007-09-07 17:53:07	S	O	1111	01/07	A1000	O	Approved		
V89A0A00008C	2007-09-07 17:53:32	S	O	1111	01/07	A1000	O	Approved		
V89A0A00008D	2007-09-07 17:54:20	S	O	1111	01/07	A1000	O	Approved		
V89A0A00008E	2007-09-07 17:56:02	S	O	1111	01/07	A1000	O	Approved		
V89A0A00008F	2007-09-07 17:56:33	S	O	1111	01/07	A1000	O	Approved		
V89A0A000090	2007-09-07 18:01:31	S	O	1111	01/07	A1000	O	Approved		
V89A0A000091	2007-09-07 18:01:44	S	O	1111	01/07	A1000	O	Approved		
V89A0A000092	2007-09-07 18:02:19	S	O	1111	01/07	A1000	O	Approved		
V89A0A000093	2007-09-07 18:03:09	S	O	1111	01/07	A1000	O	Approved		

Creating a Report Template

This sample demonstrates how to create a report template from a standard Daily Activity Report.

Creating a Report Template Request

This request creates a report template from a Daily Activity Report with a report template name of 'My Template.' It sets the values of two parameters (`name1` and `name2`). The parameter names and values are stored in the database.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <createTemplateRequest>
        <templateDefinition>
            <templateName>My Template</templateName>
            <reportName>DailyActivityReport</reportName>
                <reportParam>
                    <paramName>name1</paramName>
                    <paramValue>value1</paramValue>
                </reportParam>
                <reportParam>
                    <paramName>name2</paramName>
                    <paramValue>value2</paramValue>
                </reportParam>
            </templateDefinition>
        </createTemplateRequest>
</reportingEngineRequest>
```

NOTE: To retrieve all of your report templates, pass <getTemplateRequest/>, without specifying any child elements.

Creating a Report Template Response

In the response to creating a report template, the response code of 100 indicates that the request completed successfully. Thereafter, when 'My Template' is run, these parameters are automatically retrieved from the database and used to generate the Daily Activity Report.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
```

```
        <responseMsg>Request has completed successfully</responseMsg>
    </baseResponse>
</reportingEngineResponse>
```

Updating a Report Template

This sample demonstrates how to update the parameter information in a report template.

Updating a Report Template Request

The request specifies the report template to update by its name ‘My Template’ and the report type (Daily Activity Report) upon which it was originally based.Updating a report template is not an additive process. It replaces the parameters and values previously defined for ‘My Template’ with new parameters and values.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <updateTemplateRequest>
        <templateDefinition>
            <templateName>My Template</templateName>
            <reportName>DailyActivityReport</reportName>
            <reportParam>
                <paramName>name3</paramName>
                 <paramValue>value3</paramValue>
            </reportParam>
            <reportParam>
                <paramName>name4</paramName>
                <paramValue>value4</paramValue>
            </reportParam>
        </templateDefinition>
    </updateTemplateRequest>
</reportingEngineRequest>
```

Updating a Report Template Response

In the response to updating a report template, the response code of 100 indicates that the request completed successfully. The `getTemplateRequest` can be used to check the new parameters and values.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
```

```
    <baseResponse>
       <responseCode>100</responseCode>
       <responseMsg>Request has completed successfully</responseMsg>
    </baseResponse>
</reportingEngineResponse>
```

Getting a Report Template

This sample demonstrates how to retrieve a report template.

Getting a Report Template Request

The request specifies the report template name ‘My Template’ to retrieve.

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <getTemplateRequest>
        <templateName>My Template</templateName>
     </getTemplateRequest>
</reportingEngineRequest>
```

NOTE: To retrieve all of your report templates, pass `<getTemplateRequest/>, without specifying any child elements.`

Getting a Report Template Response

In the response to retrieving a report template, the response code of 100 indicates that the request completed successfully. The response returns the report template name, the report type (Daily Activity Report) upon which it was originally based, as well as the parameter names and values defined for this report template.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
    <baseResponse>
       responseCode>100</responseCode>
       <responseMsg>Request has completed successfully</responseMsg>
    </baseResponse>
    <getTemplateResponse>
       <templateDefinition>
          <templateName>My Template</templateName>
          <reportName>DailyActivityReport</reportName>
```

```
            <reportParam>
                <paramName>name3</paramName>
                <paramValue>value3</paramValue>
            </reportParam>
            <reportParam>
                <paramName>name4</paramName>
                <paramValue>value4</paramValue>
            </reportParam>
        </templateDefinition>
    </getTemplateResponse>
</reportingEngineResponse>
```

Deleting a Report Template

This sample demonstrates how to delete a report template.

Deleting a Report Template Request

The request to delete a report template includes the `templateName` ‘My Template.’

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
     <deleteTemplateRequest>
        <templateName>My Template</templateName>
     </deleteTemplateRequest>
</reportingEngineRequest>
```

Deleting a Report Template Response

In the response to deleting a report template, the response code of 100 indicates that the request completed successfully. If you were to specify a bogus name for the report template (in error), `responseCode` 113, “Unknown template requested.”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>113</responseCode>
        <responseMsg>Unknown template requested</responseMsg>
     </baseResponse>
</reportingEngineResponse>
</reportingEngineRequest>
```

Creating a Schedule

This sample demonstrates how to create a schedule for a report template.

Creating a Schedule Request

```
?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
   <createScheduleRequest>
        <scheduleDefinition>
            <scheduleName>My Schedule</scheduleName>
            <templateName>My Template</templateName>
            <notificationRequired>false</notificationRequired>
            <daily></daily>
        </scheduleDefinition>
     </createScheduleRequest>
</reportingEngineRequest>
```

Creating a Schedule Response

In the response to creating a schedule, the response code of 100 indicates that the request completed successfully.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
</reportingEngineResponse>
```

Getting a Schedule

Getting a Schedule Request

```
?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
    <getScheduleRequest>
        <scheduleName>My Schedule</scheduleName>
    </getScheduleRequest>
</reportingEngineRequest>
```

NOTE: To retrieve all of your scheduled reports, pass `<getScheduleRequest/>`, without specifying any child elements.

Getting a Schedule Response

In the response to getting a schedule, the response code of 100 indicates that the request completed successfully. The response returns the name of the schedule, the report template scheduled, and the schedule (daily).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
    <baseResponse>
            <responseCode>100</responseCode>
            <responseMsg>Request has completed successfully</responseMsg>
    </baseResponse>
    <getScheduleResponse>
            <scheduleDefinition>
               <scheduleName>My Schedule</scheduleName>
               <templateName>My Template</templateName>
               <notificationRequired>false</notificationRequired>
               <daily></daily>
            </scheduleDefinition>
        </getScheduleResponse>
</reportingEngineResponse>
```

Updating a Schedule

This sample demonstrates how to update a schedule.

Updating a Schedule Request

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
   <updateScheduleRequest>
        <scheduleDefinition>
            <scheduleName>My Schedule</scheduleName>
            <templateName>My Template</templateName>
            <notificationRequired>false</notificationRequired>
            <daily/>
        </scheduleDefinition>
 </updateScheduleRequest>
```

Updating a Schedule Response

In the response to updating a schedule, the response code of 100 indicates that the request completed successfully.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
</reportingEngineResponse>
```

Deleting a Schedule

This sample demonstrates how to delete a schedule.

Deleting a Schedule Request

```
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
    <deleteScheduleRequest>
        <scheduleName>My Schedule</scheduleName>
    </deleteScheduleRequest>
</reportingEngineRequest>
```

Deleting a Schedule Response

In the response to deleting a schedule, the response code of 100 indicates that the request completed successfully.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
     <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
     </baseResponse>
</reportingEngineResponse>
```

Running a Transaction ID Search

This sample demonstrates how to run a search by transaction ID.

Running a Transaction ID Search Request

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>appstest</user>
        <vendor>appstest</vendor>
        <partner>PayPal</partner>
        <password>password5</password>
    </authRequest>
 <runSearchRequest>
   <searchName>TransactionIDSearch</searchName>
   <reportParam>
       <paramName>transaction_id</paramName>
    <paramValue>V89A0A000264</paramValue>
   </reportParam>
   <pageSize>50</pageSize>
 </runSearchRequest>
</reportingEngineRequest>
```

Running a Transaction ID Search Response

In the response to running the transaction ID search, the `responseCode` of 100 indicates that the report completed successfully. The `reportId` 132 returned is used to uniquely identify this report in future calls to the Reporting API.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
        <baseResponse>
            <responseCode>100</responseCode>
            <responseMsg>Request has completed successfully</responseMsg>
        </baseResponse>
        <runSearchResponse>
            <reportId>132</reportId>
            <statusCode>3</statusCode>
            <statusMsg>Report has completed successfully</statusMsg>
        </runSearchResponse>
</reportingEngineResponse>
```

Running a Profile - Next Payment Date Report

This sample demonstrates how to run a profile - next payment date report.

Running a Profile - Next Payment Request

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
    <runReportRequest>
        <reportName>RBProfileNPDReport</reportName>
        <reportParam>
            <paramName>start_date</paramName>
            <paramValue>2011-06-22 00:00:00</paramValue>
        </reportParam>
        <reportParam>
            <paramName>end_date</paramName>
            <paramValue>2011-08-30 00:00:00</paramValue>
        </reportParam>
        <pageSize>50</pageSize>
    </runReportRequest>
</reportingEngineRequest>
```

Running a Profile - Next Payment Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
    <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
    </baseResponse>
    <runReportResponse>
        <reportId>132</reportId>
        <statusCode>3</statusCode>
        <statusMsg>Report has completed successfully</statusMsg>
    </runReportResponse>
</reportingEngineResponse>
```

Running a Profile - Profile Modified Date Report

This sample demonstrates how to run a profile - profile modified date report.

Running a Profile - Profile Modified Date Request

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
        <authRequest>
            <user>User</user>
            <vendor>Vendor</vendor>
            <partner>Partner</partner>
            <password>Password</password>
        </authRequest>
        <runReportRequest>
            <reportName>RBProfilePMDReport</reportName>
            <reportParam>
                <paramName>start_date</paramName>
                <paramValue>2010-06-22 00:00:00</paramValue>
            </reportParam>
            <reportParam>
                 <paramName>end_date</paramName>
                 <paramValue>2010-08-30 00:00:00</paramValue>
            </reportParam>
            <pageSize>50</pageSize>
        </runReportRequest>
</reportingEngineRequest>
```

Running a Profile - Profile Modified Date Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
        <baseResponse>
            <responseCode>100</responseCode>
            <responseMsg>Request has completed successfully</responseMsg>
        </baseResponse>
        <runReportResponse>
            <reportId>132</reportId>
            <statusCode>3</statusCode>
            <statusMsg>Report has completed successfully</statusMsg>
        </runReportResponse>
</reportingEngineResponse>
```

Retrieving a List of All Scheduled Reports that were Run

This sample demonstrates how to retrieve a list of report IDs for all of the scheduled reports that were run. The report ID returned can then be used for "Getting Metadata" on page 35 and "Getting Results by Report ID" on page 34.

Retrieving a List of All Scheduled Reports Request

```
<?xml version="1.0" encoding="UTF-8"?>
<reportingEngineRequest>
    <authRequest>
        <user>User</user>
        <vendor>Vendor</vendor>
        <partner>Partner</partner>
        <password>Password</password>
    </authRequest>
    <runReportRequest>
        <reportName>RetrieveReports</reportName>
        <pageSize>1000</pageSize>
    </runReportRequest>
</reportingEngineRequest>
```

Retrieving a List of All Scheduled Reports Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reportingEngineResponse>
    <baseResponse>
        <responseCode>100</responseCode>
        <responseMsg>Request has completed successfully</responseMsg>
    </baseResponse>
    <runReportResponse>
        <reportId>RE1234567890</reportId>
        <statusCode>3</statusCode>
        <statusMsg>Report has completed successfully</statusMsg>
    </runReportResponse>
</reportingEngineResponse>
```

C Report Parameters

Reports in This Appendix

This appendix describes the input and output parameters for the following reports:

- ACH Activity Summary Report
- ACH Batch ID Report
- ACH Detail Return Activity Report
- ACH Notification of Change Report
- ACH Return Activity Report
- ACH Settlement Report
- ACH Transaction Report
- Auto Credit Report
- Auto Delayed Capture Report
- Auto Resettle Report
- Auto Void Report
- Batch ID Report
- Custom Report
- Daily Activity Report
- Filter Scorecard Report
- Fraud Protection Report
- PayflowLink Orders Report
- PayPal Settlement Summary Report
- Profile Next Payment Report
- Profile Modified Date Report
- Recurring Billing Report
- Recurring Profile Failed Report
- Recurring Profile Report
- Settlement Report
- Shipping and Billing Report
- Transaction Summary Report

ACH Activity Summary Report

TABLE C.1 ACH Activity Summary Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
report_date	Date for which report needs to be pulled	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
sort_by	Column to sort data by	string	N	Transaction ID, Transaction Time, Transaction Type, Tender Type, Amount, Result	Transaction Time	
sort_option	Sort by ascending or descending order	string	N	Descending	No Order	
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:	GMT-7	
include_pending_settlement		string	N	true, false	TRUE	
include_settled		string	N	true, false	TRUE	
include_failed		string	N	true, false	TRUE	

TABLE C.2 ACH Activity Summary Report output parameters

Column name	Type	Comment
Transaction ID	string	
Time	date	
Type	string	

TABLE C.2 ACH Activity Summary Report output parameters

Tender Type	string	
Account Number	string	
Account ABA	string	
Amount	currency	
Status	string	
Description	string	
Result Code	number	
Response Msg	string	
Result Code Msg	string	
Customer Ref	string	
Currency Symbol	string	

ACH Batch ID Report

TABLE C.3 ACH Batch ID Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y			Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N		GMT-7	
host_type		number	Y	68		
batch_id		number	N			
BatchByCard		string	N	true, false	FALSE	
batch_type		string	N	Consumer Batch, Merchant Batch	Merchant Batch	

TABLE C.4 ACH Batch ID Report output parameters

Column name	Type	Comment
Host Type	number	
Batch ID	number	
Tender Type	string	If BatchByCard is given as TRUE in input
Settled Date	date	
Total Sales/Delayed Captures/Voice Authorizations (Amount)	currency	

TABLE C.4 ACH Batch ID Report output parameters

Total Sales/Delayed Captures/Voice Authorizations (Count)	number	
Total Credits (Amount)	currency	
Total Credits (Count)	number	
Total Amount	currency	
Total Count	string	
Currency Symbol	string	
Begin Time	date	
End Time	date	

ACH Detail Return Activity Report

TABLE C.5 ACH Detail Return Activity Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N		GMT-7	

TABLE C.6 ACH Detail Return Activity Report output parameters

Column name	Type	Comment
Transaction ID	string	
Time	date	
Addenda Return Code Date	date	
Type	string	
Tender Type	string	
Account Number	string	
Account ABA	string	
Amount	currency	
Status	string	
Description	string	
Result Code	number	
Response Msg	string	
Result Code Msg	string	
Customer Ref	string	

TABLE C.6 ACH Detail Return Activity Report output parameters

Batch ID	number	
Currency Symbol	string	

ACH Notification of Change Report

TABLE C.7 ACH Notification of Change Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N		GMT-7	

TABLE C.8 ACH Notification of Change Report output parameters

Column name	Type	Comment
Account ABA	string	
Account Number	string	
New Name	string	
Account Type	string	
Noc Type	string	
Deposit Date	date	
Change Information	string	

ACH Return Activity Report

TABLE C.9 Auto Return Activity Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y			Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y			Y
timezone	Timezone used to interpret times	string	N		GMT-7	

TABLE C.10 ACH Return Activity Report output parameters

Column name	Type	Comment
Status Code	string	
Description	string	
Total Amount	currency	
Total Items	number	
Currency Symbol	string	

ACH Settlement Report

TABLE C.11 ACH Settlement Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y			Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N		GMT-7	

TABLE C.12 ACH Settlement Report output parameters

Column name	Type	Comment
Transaction ID	string	
Time	date	
Type	string	
Tender Type	string	
Account Number	string	
Account ABA	string	
Amount	currency	
Status	string	
Description	string	
Result Code	number	
Response Msg	string	
Result Code Msg	string	
Customer Ref	string	
Batch ID	number	

TABLE C.12 ACH Settlement Report output parameters

Currency Symbol	string	
Comment1	string	
Comment2	string	
Fund Status	string	

ACH Transaction Report

TABLE C.13 ACH Transaction Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
report_date	Start date for which report needs to be pulled	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
sort_by		string	N	Transaction ID, Transaction Time, Transaction Type, Amount, Result	Transaction Time	
sort_option	Sort by ascending or descending	string	N	Descending	No Order	
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.14 ACH Transaction Report output parameters

Column name	Type	Comment
Transaction ID	string	
Time	date	
Type	string	

TABLE C.14 ACH Transaction Report output parameters

Tender Type	string	
Account Number	string	
Account ABA	string	
Amount	currency	
Status	string	
Description	string	
Result Code	number	
Response Msg	string	
Result Code Msg	string	
Charge Desc	string	
Customer Ref	string	
Comment1	string	
Comment2	string	
Prenote	string	
Currency Symbol	string	

Auto Credit Report

TABLE C.15 Auto Credit Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y

TABLE C.15 Auto Credit Report input parameters

timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.16 Auto Credit Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Transaction ID	string	
Time	date	
AVS Zip Match	string	
AVS Street Match	string	
Type	string	
Tender Type	string	
Amount	currency	
Result Code	number	
Comment1	string	
Comment2	string	

Auto Delayed Capture Report

TABLE C.17 Auto Delayed Capture Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y

TABLE C.17 Auto Delayed Capture Report input parameters

timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.18 Auto Delayed Capture Report output parameters

Column name	Type	Comment
Order ID	string	Only if CyberCash merchant
Transaction ID	string	
Time	date	
AVS Zip Match	string	
AVS Street Match	string	
Type	string	
Tender Type	string	
Amount	currency	
Result Code	number	
Comment1	string	
Comment2	string	

Auto Resettle Report

TABLE C.19 Auto Resettle Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.20 Auto Resettle Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Transaction ID	string	
Time	date	
AVS Zip Match	string	
AVS Street Match	string	
Type	string	
Tender Type	string	
Amount	currency	
Result Code	number	
Comment1	string	
Comment2	string	

Auto Void Report

TABLE C.21 Auto Void Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.22 Auto Void Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Transaction ID	string	
Time	date	
AVS Zip Match	string	
AVS Street Match	string	
Type	string	
Tender Type	string	
Amount	currency	
Result Code	number	
Comment1	string	
Comment2	string	

Batch ID Report

TABLE C.23 Batch ID Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
processor	Processor to report on	string	Y	PayPal	N/A	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N
group_by_tender	Should group results by tender?	string	N	true false	false	N

TABLE C.24 Batch ID Report output parameters

Column name	Type	Comment
Batch ID	number	Only if group_by_tender is set to true
Settled Date	date	
Total Sales/Delayed Captures/Voice Authorizations (Amount)	currency	
Total Sales/Delayed Captures/Voice Authorizations (Count)	number	
Total Credits (Amount)	currency	
Total Credits (Count)	number	
Total Amount	currency	
Total Count	number	
tender_type	string	Only if group-by-tender is set to true

Custom Report

TABLE C.25 Custom Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
minimum_amount	Minimum amount to report on	number	N		0	N
maximum_amount	Maximum amount to report on	number	N		10000	N
results	Results to report on	string	N	All Approvals Only Declines Only Approvals and Declines <Comma Separated List>	All	N
include_amex	Include American Express cards in report (PayPal only)	string	N	true false	true	N
include_diners	Include Diners cards in report (PayPal only)	string	N	true false	true	N

TABLE C.25 Custom Report input parameters

include_discover	Include Discover cards in report (PayPal only)	string	N	true false	true	N
include_enroute	Include Enroute cards in report (PayPal only)	string	N	true false	true	N
include_jcb	Include JCB cards in report (PayPal only)	string	N	true false	true	N
include_mastercard	Include Mastercard cards in report	string	N	true false	true	N
include_visa	Include Visa cards in report	string	N	true false	true	N
include_authorization	Include Authorizations in report	string	N	true false	true	N
include_sale	Include Sales in report	string	N	true false	true	N
include_delayedcapture	Include Delayed Captures in report	string	N	true false	true	N
include_credit	Include Credits in report	string	N	true false	true	N
include_void	Include Voids in report	string	N	true false	true	N
include_voiceauthorization	Include Voice Authorizations in report (PayPal only)	string	N	true false	true	N
include_buyer_authenticated	Include Buyer Authenticated transactions in report	string	N	true false	false	N
include_not_buyer_authenticated	Include Not Buyer Authenticated transactions in report	string	N	true false	false	N

TABLE C.25 Custom Report input parameters

recurring_only	Only include recurring transactions in report	string	N	true false	false	N
sort_by	Column to sort data by	string	N	Transaction ID Transaction Time Transaction Type Tender Type Amount Result	Transaction Time	N
sort_option	Sort by ascending or descending	string	N	Ascending Descending	Ascending	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N
show_order_id	Return Order ID column in result?	string	N	true false	true	N
show_transaction_id	Return Transaction ID column in result?	string	N	true false	true	N
show_time	Return Time column in result?	string	N	true false	true	N
show_type	Return Type column in result?	string	N	true false	true	N
show_tender_type	Return Tender Type column in result?	string	N	true false	true	N
show_account_number	Return Account Number column in result?	string	N	true false	true	N

TABLE C.25 Custom Report input parameters

show_expires	Return Expires column in result?	string	N	true false	true	N
show_aba_routing_number	Return ABA Routing Number column in result?	string	N	true false	true	N
show_amount	Return Amount column in result?	string	N	true false	true	N
show_result_code	Return Result Code column in result?	string	N	true false	true	N
show_response_msg	Return Response Msg column in result?	string	N	true false	true	N
show_comment1	Return Comment1 column in result?	string	N	true false	true	N
show_comment2	Return Comment2 column in result?	string	N	true false	true	N
show_tax_amount	Return Tax Amount column in result?	string	N	true false	true	N
show_purchase_order	Return Purchase Order column in result?	string	N	true false	true	N
show_original_transaction_id	Return Original Transaction ID column in result?	string	N	true false	true	N
show_avs_street_match	Return AVS Street Match column in result?	string	N	true false	true	N

TABLE C.25 Custom Report input parameters

show_avs_zip_match	Return AVS Zip Match column in result?	string	N	true false	true	N
show_invoice_number	Return Invoice Number column in result?	string	N	true false	true	N
show_authcode	Return Authcode column in result?	string	N	true false	true	N
show_batch_id	Return Batch ID column in result?	string	N	true false	true	N
show_csc_match	Return CSC Match column in result?	string	N	true false	true	N
show_billing_first_name	Return Billing First Name column in result?	string	N	true false	true	N
show_billing_last_name	Return Billing Last Name column in result?	string	N	true false	true	N
show_billing_company_name	Return Billing Company Name column in result?	string	N	true false	true	N
show_billing_address	Return Billing Address column in result?	string	N	true false	true	N
show_billing_city	Return Billing City column in result?	string	N	true false	true	N
show_billing_state	Return Billing State column in result?	string	N	true false	true	N
show_billing_zip	Return Billing Zip column in result?	string	N	true false	true	N

TABLE C.25 Custom Report input parameters

show_billing_email	Return Billing Email column in result?	string	N	true false	true	N
show_billing_country	Return Billing Country column in result?	string	N	true false	true	N
show_shipping_first_name	Return Shipping First Name column in result?	string	N	true false	true	N
show_shipping_last_name	Return Shipping Last Name column in result?	string	N	true false	true	N
show_shipping_address	Return Shipping Address column in result?	string	N	true false	true	N
show_shipping_city	Return Shipping City column in result?	string	N	true false	true	N
show_shipping_state	Return Shipping State column in result?	string	N	true false	true	N
show_shipping_zip	Return Shipping Zip column in result?	string	N	true false	true	N
show_shipping_country	Return Shipping Country column in result?	string	N	true false	true	N
show_customer_code	Return Customer Code column in result?	string	N	true false	true	N

TABLE C.25 Custom Report input parameters

show_freight_amount	Return Freight Amount column in result?	string	N	true false	true	N
show_duty_amount	Return Duty Amount column in result?	string	N	true false	true	N

TABLE C.26 Custom Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Transaction ID	string	
Time	date	
Type	string	
Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
ABA Routing Number	string	
Amount	currency	
Result Code	number	
Response Msg	string	
Comment1	string	
Comment2	string	
Tax Amount	currency	
Purchase Order	string	
Original Transaction ID	string	
AVS Street Match	string	
AVS Zip Match	string	
Invoice Number	string	
Authcode	string	
Batch ID	number	

TABLE C.26 Custom Report output parameters

CSC Match	string	
Billing First Name	string	
Billing Last Name	string	
Billing Company Name	string	
Billing Address	string	
Billing City	string	
Billing State	string	
Billing Zip	string	
Billing Email	string	
Billing Country	string	
Shipping First Name	string	
Shipping Last Name	string	
Shipping Address	string	
Shipping City	string	
Shipping State	string	
Shipping Zip	string	
Shipping Country	string	
Customer Code	string	
Freight Amount	currency	
Duty Amount	currency	

Daily Activity Report

TABLE C.27 Daily Activity Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
report_date	Date to run report for	date YYYY-MM-DD	Y		N/A	Y
sort_by	Column to sort data by	string	N	Transaction ID Transaction Time Transaction Type Tender Type Amount Result	Trans Time	N
sort_option	Sort by ascending or descending	string	N	Ascending Descending	Ascending	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

Currency values are returned in the three-character currency code specified. The values are:

- USD (US dollar) (default value)
- EUR (Euro)
- GBP (UK pound)
- CAD (Canadian dollar)
- JPY (Japanese Yen)
- AUD (Australian dollar)

TABLE C.28 Daily Activity Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Transaction ID	string	
Time	date	
Type	string	
Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Result	number	
Response Msg	string	
Comment1	string	
Comment2	string	
Client IP	string	
Authcode	string	
AVS Street Match	string	
AVS Zip Match	string	

Filter Scorecard Report

TABLE C.29 Filter Scorecard Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.30 Filter Scorecard Report output parameters

Column name	Type
Filter	string
Times Triggered	number
Trigger Percentage	number

Fraud Protection Report

TABLE C.31 Fraud Protection Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
transaction_type	Transaction types to report on	string	Y	Review Reject Accept Screened By Filters Not Screened By Filters	N/A	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.32 Fraud Protection Report output parameters

Column name	Type
Transaction ID	string
Time	date
Type	string
Tender Type	string
Amount	currency
Deployment Mode	string
Fraud Transaction	string

Payflow Link Orders Report

TABLE C.33 PayflowLink Orders Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.34 PayflowLink Orders Report output parameters

Column name	Type
Time	date
Transaction ID	string
Billing Name	string
Billing Address	string
Billing City	string
Billing State	string
Billing Zip	string
Billing Country	string
Billing Phone	string
Billing Fax	string
Billing Email	string
Shipping Name	string
Shipping Address	string

TABLE C.34 PayflowLink Orders Report output parameters

Shipping City	string
Shipping State	string
Shipping Zip	string
Shipping Country	string
Shipping Phone	string
Shipping Fax	string
Shipping Email	string
Amount	currency
Tax Amount	currency
Ship Amount	currency
Description	string
Type	string
Result Code	number

PayPal Settlement Summary Report

TABLE C.35 PayPal Settlement Summary Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.36 PayPal Settlement Summary Report output parameters

Column name	Type	Comment
Date	date	
Sales and Delay Captures Number	number	
Sales and Delay Captures Amount	currency	Values are returned in the specified, supported currency.
Credits Number	number	
Credits Amount	currency	
Fees Amount	currency	
Net Amount	currency	
Currency Symbol	string	

Profile Next Payment Report

TABLE C.37 Profile Next Payment Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime - YYYY-MM-DD HH:MI:SS	Y	Current and future date/time values	N/A	Y
end_date	End date to run report for	datetime - YYYY-MM-DD HH:MI:SS	Y	Current and future date/time values	N/A	Y

TABLE C.38 Profile Next Payment Report output parameters

Column name	Type
Profile Id	number
Profile Name	string
Last Name	string
Status	number
Account Number	string - The last 4 digits are returned
Expires	string
Amount	currency
Established Date	date
Start Date	date
Frequency	string
Terms	number
Next Payment Date	date
Deactivation Date	date
Profile Modified Date	date

Profile Modified Date Report

TABLE C.39 Profile Next Payment Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime - YYYY-MM-DD HH:MI:SS	Y	Current and future date/time values	N/A	Y
end_date	End date to run report for	datetime - YYYY-MM-DD HH:MI:SS	Y	Current and future date/time values	N/A	Y

TABLE C.40 Profile Modified Date Report output parameters

Column name	Type
Profile Id	number
Profile Name	string
Last Name	string
Status	number
Account Number	string - The last 4 digits are returned
Expires	string
Amount	currency
Established Date	date
Start Date	date
Frequency	string
Terms	number
Next Payment Date	date
Deactivation Date	date
Profile Modified Date	date

Recurring Billing Report

TABLE C.41 Recurring Billing Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime - YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime - YYYY-MM-DD HH:MI:SS	Y		N/A	Y
include_ approvals	Incl approvals in report	string	N	true false	true	N
include_ declines	Include declines in report	string	N	true false	true	N
timezone	Timezone to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N
tender_ type	Run report for ACH or Credit	string	N	ACH Credit	Credit	N

TABLE C.42 Recurring Billing Report output parameters

Column name	Type
Profile Name	string
Profile ID	string
Status	string
Payment Number	number
Retry Number	number
Time	date
Retry Reason	string
Retry Flag	number
Transaction ID	string
Result Code	number

Recurring Profile Failed Report

TABLE C.43 Recurring Profile Failed Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	date YYYY-MM-DD	Y		N/A	Y
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.44 Recurring Profile Failed Report output parameters

Column name	Type
Profile Name	string
Profile ID	string
Status	string
Payment Number	number
Retry Number	number
Time	date
Retry Reason	string
Retry Flag	number
Transaction ID	string
Result Code	number

Recurring Profile Report

TABLE C.45 Recurring Profile Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
profile_id	Profile to report on	string	Y		N/A	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N
tender_type	Run report for ACH or Credit	string	N	ACH Credit	Credit	N

TABLE C.46 Recurring Profile Report output parameters

Column name	Type
Profile Name	string
Profile ID	string
Status	string
Payment Number	number
Retry Number	number
Time	date
Retry Reason	string
Retry Flag	number
Transaction ID	string
Result Code	number

Settlement Report

TABLE C.47 Settlement Report input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
processor	Processor to report on	string	Y	Name of the processor for which the settlement report is requested	N/A	N
settlement_status	Settled or non-settled transactions	string	N	Settled Unsettled All	All	N
transaction_type	Transaction types to report on	string	N	Sales Credits All	All	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.48 Settlement Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Transaction ID	string	
Time	date	
Type	string	

TABLE C.48 Settlement Report output parameters

Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Result Code	number	
Response Msg	string	
Comment1	string	
Comment2	string	
Batch ID	number	
Currency Symbol	string	Alphabetic currency symbol, for example USD
PayPal Transaction ID	string	Returned only if merchant has a PayPal Internet Merchant Account
PayPal Fees	currency	Returned only if merchant has a PayPal Internet Merchant Account
PayPal Email ID	string	Buyer's PayPal email address ; returned only if merchant has a PayPal Internet Merchant Account

Shipping and Billing Report

TABLE C.49 Shipping and Billing Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
report_date	Date to run report for	date YYYY-MM-DD	Y		N/A	Y
sort_by	Column to sort data by	string	N	Transaction ID Transaction Time Transaction Type Tender Type Amount Result	Trans Time	N
sort_option	Sort by ascending or descending	string	N	Ascending Descending	Ascending	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.50 Shipping and Billing Report output parameters

Column name	Type	Comment
Order ID	string	Only if Cybercash merchant
Time	date	
Transaction ID	string	
Captured	string	
Type	string	
Billing First Name	string	
Billing Last Name	string	
Billing Company Name	string	
Billing Address	string	
Billing City	string	

TABLE C.50 Shipping and Billing Report output parameters

Billing State	string	
Billing Zip	string	
Billing Email	string	
Billing Country	string	
Shipping First Name	string	
Shipping Last Name	string	
Shipping Address	string	
Shipping City	string	
Shipping State	string	
Shipping Zip	string	
Shipping Country	string	
Purchase Order	string	
Customer Code	string	
Tax Amount	currency	
Freight Amount	currency	
Duty Amount	currency	
Total Amount	currency	

Transaction Summary Report

TABLE C.51 Transaction Summary Report input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
start_date	Start date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
end_date	End date to run report for	datetime YYYY-MM-DD HH:MI:SS	Y		N/A	Y
settlement_status	Settled or non-settled transactions	string	N	Settled Unsettled All	All	N
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE C.52 Transaction Summary Report output parameters

Column name	Type
Transaction Type	string
Processor	string
Trans Count	number
Result Code	number
Tender Type	string
Min Amount	currency
Max Amount	currency
Average	currency
Total Amount	currency

D Search Parameters

Transaction ID Search

TABLE D.1 Transaction ID Search input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
transaction_id	Transaction ID to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.2 Transaction ID Search output parameters

Column name	Type	Comment
Order ID	string	
Transaction ID	string	
Merchant	string	
User Name	string	
Time	string	
Type	string	
Duration	number	
Tender Type	string	
Client IP Address	string	
Account Number	string	The last 4 digits are returned
Client Version	string	
Expires	string	
Amount	currency	

TABLE D.2 Transaction ID Search output parameters

Comment1	string	
Comment2	string	
Billing First Name	string	
Billing Last Name	string	
Billing Address	string	
Billing City	string	
Billing State	string	
Billing Zip	string	
Billing Country	string	
Billing Email	string	
Shipping First Name	string	
Shipping Last Name	string	
Shipping Address	string	
Shipping City	string	
Shipping State	string	
Shipping Zip	string	
Shipping Country	string	
Recurring	string	
Result Code	number	
Response Msg	string	
Authcode	string	
Original Transaction ID	string	
AVS Street Match	string	
Original Amount	currency	
AVS Zip Match	string	
International AVS Indicator	string	
CSC Match	string	
Batch ID	number	

Batch ID Search

TABLE D.3 Batch ID Search input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
batch_id	Batch ID to search for	string	Y			
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.4 Batch ID Search output parameters

Column name	Type	Comment
Order ID	string	Only if CyberCash merchant
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

AccountNumberSearch

TABLE D.5 AccountNumberSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
account_number	Account Number to search for	string	Y			
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.6 AccountNumberSearch output parameters

Column name	Type	Comment
Order ID	string	Only if CyberCash merchant
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

CommentSearch

TABLE D.7 CommentSearch input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
comment1	Comment1 to search for	string	Y (One or both of comment1 and comment2 are required)	comment1	Comment1 to search for	string
comment2	Comment2 to search for	string	Y (One or both of comment1 and comment2 are required)	comment2	Comment2 to search for	string
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y	start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y	end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.8 CommentSearch output parameters

Column name	Type	Comment
Order ID	string	Only if CyberCash merchant
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	

TABLE D.8 CommentSearch output parameters

Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

AccountNumberRefSearch

TABLE D.9 AccountNumberRefSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
transaction_id	Transaction ID to search for	string	Y			
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.10 AccountNumberRefSearch output parameters

Column name	Type	Comment
Order ID	string	Only if CyberCash merchant
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

PurchaseOrderSearch

TABLE D.11 PurchaseOrderSearch input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
purchase_order_num	Purchase Order Number to search for	string	Y			
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.12 PurchaseOrderSearch output parameters

Column name	Type	Comment
Order ID	string	Only if CyberCash merchant
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

RecurringBillingProfileIDSearch

TABLE D.13 RecurringBillingProfileIDSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
profile_id	Profile ID to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.14 RecurringBillingProfileIDSearch output parameters

Column name	Type	Comment
Profile ID	string	
Profile Name	string	
Name	string	
Address	string	
Phone Number	string	
EMail	string	
Recurring Amount	currency	
Original Start Date	date	
Term	number	
Frequency	string	
Status	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Comment	string	
Start Date	date	
Next Payment Date	date	
Last Date	date	

TABLE D.14 RecurringBillingProfileIDSearch output parameters

Date Established	date	

RecurringBillingProfileNameSearch

TABLE D.15 RecurringBillingProfileNameSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
profile_name	Profile Name to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.16 RecurringBillingProfileNameSearch output parameters

Column name	Type	Comment
Profile Name	string	
Profile ID	string	
Status	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Frequency	string	
Start Date	date	
Date Established	date	

RecurringBillingAccountNumberSearch

TABLE D.17 RecurringBillingAccountNumberSearch input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
account_number	Account number to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.18 RecurringBillingAccountNumberSearch output parameters

Column name	Type	Comment
Profile Name	string	
Profile ID	string	
Status	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Frequency	string	
Start Date	date	
Date Established	date	

RecurringBillingCommentSearch

TABLE D.19 RecurringBillingCommentSearch input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
comment	Comment to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.20 RecurringBillingCommentSearch output parameters

Column name	Type	Comment
Profile Name	string	
Profile ID	string	
Status	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Frequency	string	
Start Date	date	
Date Established	date	

RecurringBillingAmountSearch

TABLE D.21 RecurringBillingAmountSearch input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
amount	Amount to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.22 RecurringBillingAmountSearch output parameters

Column name	Type	Comment
Profile Name	string	
Profile ID	string	
Status	string	
Account Number	string	The last 4 digits are returned
Expires	string	
Amount	currency	
Frequency	string	
Start Date	date	
Date Established	date	

FraudTransactionSearch

TABLE D.23 FraudTransactionSearch input parameters

Name	Description	Type	Required?	Allowed values	Default value	Auto-computed by scheduler?
fraud_transaction_id	transaction ID to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.24 FraudTransactionSearch output parameters

Column name	Type	Comment
Transaction ID	string	
Time	string	
Return Code Message	string	
Customer Name	string	
Email Address	string	
Phone Number	string	
Amount	currency	
Total Items	number	
SKU	string	
Account Number	string	The last 4 digits are returned
AVS Street Match	string	
AVS Zip Match	string	
CSC Match	string	
International AVS Indicator	string	
Buyer Authentication	string	
Billing Name	string	
Billing Street1	string	

TABLE D.24 FraudTransactionSearch output parameters

Billing Street2	string	
Billing City	string	
Billing State	string	
Billing Zip	string	
Billing Country	string	
Shipping Name	string	
Shipping Street1	string	
Shipping Street2	string	
Shipping City	string	
Shipping State	string	
Shipping Zip	string	
Shipping Country	string	
Client IP Address	string	
Num Custom Rules	number	
Custom Rules	string	
Num Triggered Rules	number	
Triggered Rules	string	
Num Unprocessed Rules	number	
Unprocessed Rules	string	

OrderIDSearch

TABLE D.25 OrderIDSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
order_id	Order ID to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.26 OrderIDSearch output parameters

Column name	Comment	
Order ID	string	
Transaction ID	string	
Merchant	string	
Time	string	
Type	string	
Duration	number	
Tender Type	string	
Client IP Address	string	
Account Number	string	The last 4 digits are returned
Client Version	string	
Expires	string	
Street	string	
Amount	currency	
Zip	string	
Email	string	
Name On Card	string	
Comment1	string	

TABLE D.26 OrderIDSearch output parameters

Comment2	string	
Result Code	number	
Response Msg	string	
Authcode	string	
Original Transaction ID	string	
AVS Street Match	string	
Original Amount	currency	
AVS Zip Match	string	
Batch ID	number	
AVS Result Code	string	

PayPalTransactionIDSearch

TABLE D.27 PayPalTransactionIDSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
paypal_transaction_id	PayPal transaction ID to search for	string	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.28 PayPalTransactionIDSearch output parameters

Column name	Type	Comment
PayPal Transaction ID	**string**	**Unique transaction ID of the payment returned in a DoExpressCheckoutPayment response**
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Email	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

PayPalEmailSearch

TABLE D.29 PayPalEmailSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
email	Email address to search for	string	Y			
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.30 PayPalEmailSearch output parameters

Column name	Type	Comment
PayPal Transaction ID	string	
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Email	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

PayPalNameSearch

TABLE D.31 PayPalNameSearch input parameters

Name	Description	Type	Required ?	Allowed values	Default value	Auto-computed by scheduler?
last_name	Last name to search for	string	Y			
first_name	First name to search for	string	N			
start_date	Start date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
end_date	End date to search for	datetime - YYYY-MM-DD HH:MI:SS	Y			
timezone	Timezone used to interpret times	string	N	GMT GMT+/-XX:XX	GMT	N

TABLE D.32 PayPalNameSearch output parameters

Column name	Type	Comment
PayPal Transaction ID	string	Unique transaction ID of the payment returned in a DoExpressCheckoutPayment response
Transaction ID	string	
Time	string	
Type	string	
Tender Type	string	
Email	string	
Amount	currency	
Response Msg	string	
Comment1	string	
Comment2	string	

Index

A

ACH 62
ACH Activity Summary Report 62
ACH Batch ID Report 64
ACH Detail Return Activity Report 66
ACH Notification of Change Report 67
ACH Return Activity Report 68
ACH Settlement Report 69
ACH Transaction Report 70
Auto Credit Report 71
Auto Delayed Capture Report 72
Auto Resettle Report 74
Auto Void Report 75

B

Batch ID Report 76

C

createScheduleRequest 18
createTemplateRequest 16
creating report templates 49
creating schedules 53
creating templates 49
Custom Report 77

D

Daily Activity Report 31, 85
deleteScheduleRequest 19
deleteTemplateRequest 17
deleting report templates 52
deleting templates 52, 55, 56

F

Filter Scorecard Report 87
Fraud Protection Report 88

G

getDataRequest 15, 19
getMetaDataRequest 15, 16
getResultsRequest 16
getScheduleRequest 18
getting data 38
getting metadata 35
getting report templates 51
getting results by report ID 34
getting schedules 54

M

managing report templates 16

O

on demand 11

P

Partner Manager
  overview 8
Payflow Link Orders Report 89
Payflow Pro 11
PayPal Settlement Summary Report 91
performing searches 20
Profile Modified Date Report 93
Profile Next Payment Report 92

R

Recurring Billing Report 94
Recurring Profile Failed Report 95
Recurring Profile Report 96
response codes 21
retrieve report results by reportId 34
retrieving metadata 35
retrieving report data 38
retrieving templates 51
running an ad hoc report 33
running reports 15

running reports offline 16
running transaction ID search 57
runReportRequest 15, 17
runSearchRequest 20

S

Settlement Report 97
Shipping and Billing Report 99
status codes 22

T

Transaction Summary Report 33, 101

U

updateScheduleRequest 19
updateTemplateRequest 17
updating report templates 50
updating schedules 55
updating templates 50
using report templates 17